

# MIPP Online Release Policy

August 30, 2004

# Contents

<b>1</b>	<b>Motivation for a strict policy</b>	<b>3</b>
<b>2</b>	<b>Requirements</b>	<b>3</b>
2.1	Code and build system . . . . .	3
2.2	Releases . . . . .	4
<b>3</b>	<b>Rules</b>	<b>5</b>
3.1	Code and build system . . . . .	5
3.1.1	Online will use ... for build system . . . . .	5
3.1.2	No branching of code in CVS . . . . .	5
3.2	Release tagging . . . . .	5
3.2.1	A group of people will be responsible for tagging stable releases . . . . .	5
3.2.2	Development version will be extensively tested . . . . .	5
3.2.3	Previous releases remain on disk . . . . .	5
3.2.4	Releases are static . . . . .	5
3.2.5	Detector readouts will be running the same release as mdd . . . . .	5
3.2.6	Release of each mdd will be logged into database . . . . .	5
3.2.7	Change of current release will be inserted into the log-book . . . . .	6

# 1 Motivation for a strict policy

As of August 2004, there are a number of problems with the DAQ software installed on e907daq and VME PowerPC's

1. Source code is not carefully controlled and a clear link between source code, executables and run number is missing.
2. Changes to the installed software are not announced and poorly documented.
3. It is not easy to roll back to an earlier version of DAQ, that is the steps can only be done by an expert.

Potential consequences of this lax situation include

- Increased downtime
- Frustration from running incompatible versions of code

## 2 Requirements

In order for MIPP to smoothly take data, we find it necessary to impose the following requirements on the development of MIPP online software and production of software releases that are known to be working.

### 2.1 Code and build system

- Build/version system has to work on the PowerPC's and on modern UNIX systems (including e907daq)
- Development code committed into CVS should at the very least compile. Preferably, the code shouldn't crash when executed.
- Developers should work off the latest release. If a given code does not compile or does not work, the developer responsible for the introduction of the bug should be notified. If the person who found the bug knows how to fix it quickly, (s)he should do that.
- Developer should be able to test a change off a tagged release.

## 2.2 Releases

*Italised statements should probably be taken out.*

- *All ppcs and e907 run executables from the same release.*
- Run numbers must be associated with the release that produced the data.
- There must be a simple way for shifters to use an earlier release of the code.
- A number of past production versions must remain compiled on disk.
- Versioning system should not be confusing to developers.
- Entire release must go through rigorous testing before it is approved to be called a production release. This includes all code inside `mdd`, `RunControl`, `Helpers`, `e907r2dm`, and `Detectors` CVS directories.
- Upon succesful completion of tests the process of approval should be simple, but should involve more than one person.
- A summary of changes for a new production release is recorded.
- All of online software gets tagged at the same time (i.e. code running exclusively on PPC's AND exclusively on e907daq).
- Collaboration-wide notification of a DAQ change. *Is logbook enough?*
- Backward-incompatible DAQ changes are announced and agreed upon before the new release is used on e907daq.
- Production releases are static. No changes are allowed in that release. A new release must be created for any change.

## **3 Rules**

### **3.1 Code and build system**

#### **3.1.1 Online will use ... for build system**

#### **3.1.2 No branching of code in CVS**

Code should either be considered useful and useable, and therefore maintained, OR it should be rewritten. If it is determined that old code should be revived, it should appear as a new executable or class/library or should overwrite the HEAD version in CVS.

### **3.2 Release tagging**

#### **3.2.1 A group of people will be responsible for tagging stable releases**

All the members of the group will agree that the release is ready to be used for production data before it is used by shifters.

#### **3.2.2 Tagging will be the only supported versioning system**

#### **3.2.3 All packages used by online will receive the same tag**

Even if only one package was changed, all code will get tagged.

#### **3.2.4 Development version will be extensively tested**

Testing should not be limited to testing changes. The following features must be successfully tested:

- Start & stop runs

#### **3.2.5 Previous releases remain on disk**

#### **3.2.6 Releases are static**

A buggy release will be fixed, tagged as a new release and removed from further usage.

**3.2.7 Detector readouts will be running the same release as mdd**

**3.2.8 Release of each mdd will be logged into database**

Given rule 3.2.5, the release tag of each of the detector readouts will be defined.

**3.2.9 Change of current release will be inserted into the logbook**

Documentation of differences to previous release will be inserted as well. *Can we agree to write meaningful CVS log comments so that a summary of all comments would be enough?*